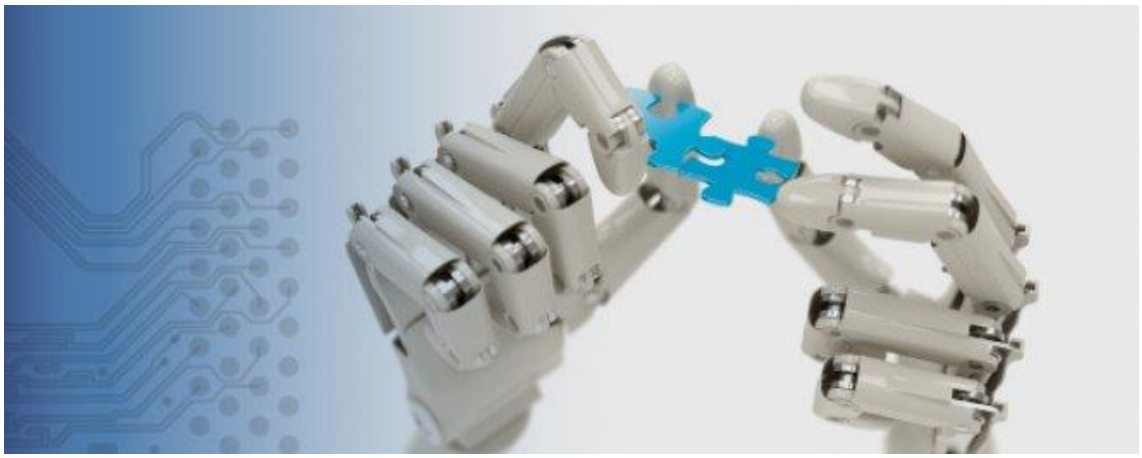
ROBOTICS MASTER

Universitat de Vic



Subject: Robotics Integration

Unit: 1_B: Software Development Environment

Exercixse 1.4: CMake

Author: Toni Guasch Serra

Date: 2015-10-27

CMake. Exercise 1.4

```
sudo apt-get install libopencv-dev
```

a. Once logged in your git account, go to:
   https://github.com/beta-robots/webcam_capture.git

   and find the button "**fork**". Click it!. This action "forks" this repository to your git space, so it creates a new repository in your git space with the same content.

b. Clone **YOUR** recently created (forked) repository:

```
git clone https://github.com/my_github_name/webcam_capture.git
```

c. Try to understand both the code and the CMakeLists.txt

d. Build the code and execute it with your webcam

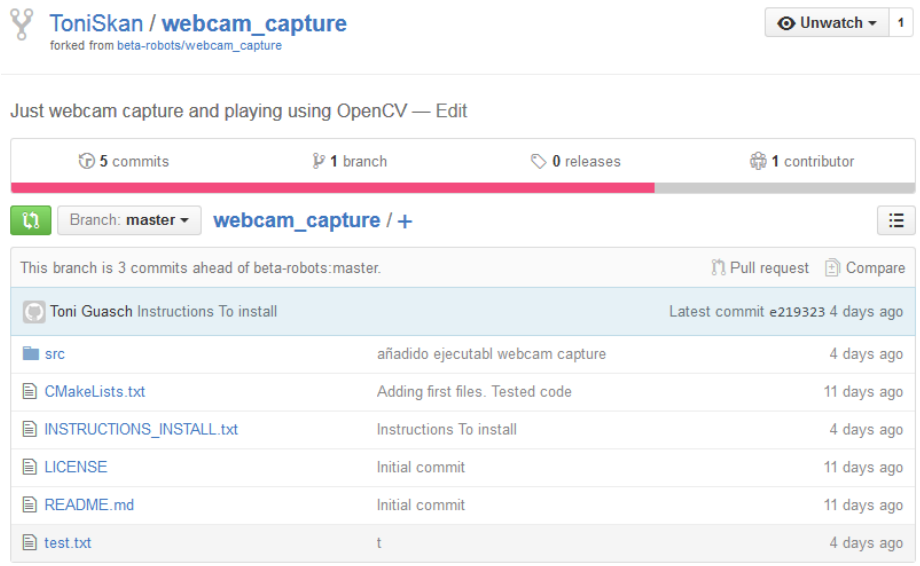
e. Edit the wiki page of the project to document the procedure to follow by a newby user who wants to download and execute your project.

f. Send a mail/alert to the professor through moodle when you consider the project is "finished"

a. Once logged in your git account, go to:

https://github.com/beta-robots/webcam_capture.git and find the button "fork". Click it!. This action "forks" this repository to your git space, so it creates a new repository in your git space with the same content.

Done:



b. Clone YOUR recently created (forked) repository:

Done in Terminal Mode in Ubuntu:

```
toni@ubuntu: ~/Documents/GIT/Webcam
toni@ubuntu:~$ cd Documents/GIT
toni@ubuntu:~/Documents/GIT$ git mkdir Webcam && cd Webcam
git: 'mkdir' is not a git command. See 'git --help'.
toni@ubuntu:~/Documents/GIT$ mkdir Webcam && cd Webcam
toni@ubuntu:~/Documents/GIT/Webcam$ git clone https://github.com/ToniSkan/webcam
_capture.git
Cloning into 'webcam_capture'...
remote: Counting objects: 19, done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 19 (delta 2), reused 0 (delta 0), pack-reused 9
Unpacking objects: 100% (19/19), done.
Checking connectivity... done.
toni@ubuntu:~/Documents/GIT/Webcam$
```

c. Try to understand both the code and the CMakeLists.txt

CMakeLists.txt (~/Documents/GIT/Webcam/webcam_capture) - gedit

Open Save Undo

CMakeLists.txt

```
#indicate minimum version
CMAKE_MINIMUM_REQUIRED(VERSION 2.6)

#project name
PROJECT(webcam_capture)

#find required packages (look for the package, usually at /usr/share/
cmake-2.8/Modules/ or /usr/share/)
FIND_PACKAGE(OpenCV REQUIRED)

#set header directories
INCLUDE_DIRECTORIES(${OpenCV_INCLUDE_DIR})

#Create an executable
ADD_EXECUTABLE(${PROJECT_NAME} src/webcam_capture.cpp)

#Link with libraries
TARGET_LINK_LIBRARIES(${PROJECT_NAME} ${OpenCV_LIBS})

# Setting this prefix will be used by INSTALL commands in next
CMakeLists
SET(CMAKE_INSTALL_PREFIX /usr/local)

#install
INSTALL(TARGETS ${PROJECT_NAME} RUNTIME DESTINATION bin)
```

```
#include "cv.h"
#include "highgui.h"
#include <iostream>
#include <cstdlib>

int main(int argc, char *argv[])
{
        //OpenCV video capture object
    cv::VideoCapture camera;

        //OpenCV image object
    cv::Mat image;

        //camera id . Associated to device number in /dev/videoX
        int cam_id;

        //check user args
        switch(argc)
        {
                case 1: //no argument provided, so try /dev/video0
                        cam_id = 0;
                        break;
                case 2: //an argument is provided. Get it and set
cam_id
                        cam_id = atoi(argv[1]);
                        break;
                default:
                        std::cout << "Invalid number of arguments.
Call program as: webcam_capture [video_device_id]. " << std::endl;
                        std::cout << "EXIT program." << std::endl;
                        break;
        }

        //advertising to the user
        std::cout << "Opening video device " << cam_id << std::endl;

    //open the video stream and make sure it's opened
    if( !camera.open(cam_id) )
        {
        std::cout << "Error opening the camera. May be invalid device
id. EXIT program." << std::endl;
        return -1;
    }


    }

    //capture loop. Out of user press a key
    while(1)
        {
                //Read image and check it
        if(!camera.read(image))
                {
            std::cout << "No frame" << std::endl;
            cv::waitKey();
        }

        //show image in a window
        cv::imshow("Output Window", image);

                //print image dimensions
                std::cout << "image size is: " << image.rows << "x" <<
image.cols << std::endl;

                //Waits 1 millisecond to check if a key has been
pressed. If so, breaks the loop. Otherwise continues.
        if(cv::waitKey(1) >= 0) break;
    }
}
```

d. Build the code and execute it with your webcam

The code was build following these instructions, without modyfing anything of CMakeLists.txt and webcam_capture.cpp

Commands to build and execute

```
Assuming at hello_world_folder there are CMakeLists.txt and hello_world.cpp :

$ cd hello_world_folder
$ mkdir build
$ cd build
$ cmake ..
$ make
$ ./hello_wolrd
```

Changing “hello_world_folder” for “src” and “./hello_wolrd” for “./webcam_capture”

When the code was build. The camera does not start.

“Error opening the camera. May be invalid device id. EXIT program”

This error could be because I’m using a virtual machine to use Ubuntu. Is possible that the virtual machine doesn’t have access to webcam resource.

```
toni@ubuntu:~/Documents/GIT/Webcam/webcam_capture/src$ make
Scanning dependencies of target webcam_capture
[100%] Building CXX object CMakeFiles/webcam_capture.dir/webcam_capture.cpp.o
Linking CXX executable webcam_capture
[100%] Built target webcam_capture
toni@ubuntu:~/Documents/GIT/Webcam/webcam_capture/src$ ./webcam_capture
Opening video device 0
Error opening the camera. May be invalid device id. EXIT program.
toni@ubuntu:~/Documents/GIT/Webcam/webcam_capture/src$
```

e. Edit the wiki page of the project to document the procedure to follow by a new user who wants to download and execute your project.

https://github.com/ToniSkan/webcam_capture/wiki

Home

Toni Guasch Serra edited this page just now · 3 revisions

Welcome to the webcam_capture wiki!

INSTRUCTIONS TO INSTALL: Before to apply this steps you must have installed CMake and OpenCV.

a.Once logged in your git account, go to: https://github.com/ToniSkan/webcam_capture.git and find the button “fork”. Click it!. This action “forks” this repository to your git space, so it creates a new repository in your git space with the same content.

b.Clone YOUR recently created (forked) repository: git clone https://github.com /my_github_name/webcam_capture.git

c.open in your terminal de directory which contain CMakeLists.txt

d.execute in your terminal: "cmake .."

e.execute in your terminal: make

f.Open the generated executable "./webcam_capture"

Then you build the executable to see the image of your Webcam

+ Add a custom footer

f. Send a mail/alert to the professor through moodle when you consider the project is “finished”

Sent this article by email